

CLAIMS

1. A method comprising:
calling a remote object that resides on a server using an object-oriented network protocol;
receiving a reply from the server, the reply containing server state information; and
caching the server state information for use in subsequent communication with the server.

2. A method as recited in claim 1, wherein the server state information is embodied as a data object.

3. A method as recited in claim 1, wherein the server state information comprises a service ID and data.

4. A method as recited in claim 1, wherein the object-oriented network protocol comprises a remote procedure call (RPC) protocol.

5. A method as recited in claim 1, further comprising:
subsequently calling the remote object using the object-oriented network protocol; and
submitting the server state information to the remote object.

1 **6.** A method comprising:
2 receiving a request for a local program object from a remote application
3 program interface on a requesting computer;
4 creating a state-caching object that contains state information pertaining to
5 the request;
6 processing the request to generate a reply; and
7 returning the reply together with the state-caching object to the requesting
8 computer.

9
10 **7.** A method as recited in claim 6, wherein the state-caching object
11 contains a service ID field to identify a computer or group of computers that
12 created the state-caching object and a data field.

13
14 **8.** A method as recited in claim 6, further comprising inserting the state-
15 caching object into the process state used to process the request.

16
17 **9.** A method as recited in claim 6, further comprising:
18 subsequently receiving another request from the requesting computer, along
19 with the state-caching object; and
20 using the state-caching object to recall state information pertaining to a
21 previous connection with the requesting computer.

22
23 **10.** A method comprising:
24 submitting a request to a server using a non-HTTP protocol over a network;
25

1 receiving a reply from the server, the reply containing a state-caching
2 object with session state information; and
3 storing the state-caching object for use in subsequent communication with
4 the server.

5
6 **11.** A method as recited in claim 10, wherein the state-caching object
7 comprises a service ID and data.

8
9 **12.** A method as recited in claim 10, wherein the non-HTTP protocol
10 comprises an object-oriented network protocol.

11
12 **13.** A method as recited in claim 10, wherein the non-HTTP protocol
13 comprises a remote procedure call (RPC) protocol.

14
15 **14.** A method as recited in claim 10, further comprising:
16 submitting a subsequent request to the server; and
17 sending the state-caching object along with the subsequent request.

18
19 **15.** A method comprising:
20 routing a request from a first computer to a second computer via a network;
21 routing a reply from the second computer back to the first computer via the
22 network, the reply carrying state information of the second computer that pertains
23 to the request; and
24 maintaining the state information within the network.
25

1 **16.** A method as recited in claim 15, wherein the server state
2 information is embodied as a data object.

3
4 **17.** A method as recited in claim 15, wherein the network comprises one
5 or more network components and the maintaining comprises storing the state
6 information of the second computer on one of the network components.

7
8 **18.** A method as recited in claim 15, wherein the network comprises
9 multiple network components and the maintaining comprises continually passing
10 the state information among the network components.

11
12 **19.** A method as recited in claim 15, wherein the network supports
13 remote procedure call protocol and the routing a request comprises passing the
14 request as part of a call to a program object located on the second computer.

15
16 **20.** A method comprising:
17 performing request/reply exchanges among multiple computers organized
18 in a computer cluster;
19 generating state-caching objects that contain state information of
20 corresponding computers as part of the request/reply exchanges;
21 storing the state-caching objects on one or more different computers within
22 the computer cluster to maintain the state information remotely from the
23 corresponding computers from which the state-caching objects originated and
24 preserve the state information in an event that one of the corresponding computers
25 fails.

1
2 **21.** A method as recited in claim 20, wherein the performing
3 request/reply exchanges comprises conducting remote procedure calls to remote
4 program objects on another computer.

5
6 **22.** A method as recited in claim 20, further comprising, in an event that
7 one of the corresponding computers fails, using the state-caching object associated
8 with the failed computer to at least partially restore state information for the failed
9 computer during recovery.

10
11 **23.** A method as recited in claim 20, wherein each state-caching object
12 contains a service ID field to identify a service as represented by one or more
13 computers that created the state-caching object and a data field.

14
15 **24.** A stateless distributed computer architecture, comprising:
16 a program object resident at a first computing device;
17 an application program interface (API) resident at a second computing
18 device to facilitate calls to the program object at the first computing device using
19 an object-oriented network protocol;

20 the program object, responsive to a call, returning a reply with a state-
21 caching object that contains state information pertaining to the first computing
22 device; and

23 wherein the state-caching object is stored on the second computing device
24 for later communication with the first computing device.
25

1 **25.** A stateless distributed computer architecture as recited in claim 24,
2 wherein the object-oriented network protocol comprises a remote procedure call
3 (RPC) protocol.

4
5 **26.** A stateless distributed computer architecture as recited in claim 24,
6 wherein the first and second computer are organized in a cluster of computers.

7
8 **27.** A method as recited in claim 24, wherein the state-caching object
9 contains a service ID field to identify the service of the first computing device and
10 a data field.

11
12 **28.** A stateless distributed computer system, comprising:
13 a network having one or more network components to route requests from a
14 first endpoint device to a second endpoint device and to route replies from the
15 second endpoint device back to the first endpoint device, wherein at least one
16 reply contains state information pertaining to the second endpoint device; and
17 the network being configured to maintain the state information and to
18 reassociate the state information with a subsequent request from the first endpoint
19 device to the second endpoint device.

20
21 **29.** A stateless distributed computer system as recited in claim 28,
22 wherein at least one of the network components stores the state information.

1 **30.** A stateless distributed computer system as recited in claim 28,
2 wherein multiple network components continually route the state information
3 amongst themselves to preserve the state information.
4

5 **31.** A computer-readable medium comprising computer-executable
6 instructions that, when executed on one or more processors, direct a computing
7 device to:

8 call a remote object that resides on a remote computer using an object-
9 oriented network protocol;

10 receive a reply from the remote computer, the reply containing state
11 information of the remote computer; and

12 cache the state information for use in subsequent communication with the
13 remote computer.
14

15 **32.** A computer-readable medium as recited in claim 31, wherein the
16 state information is embodied as a data object.
17

18 **33.** A computer-readable medium as recited in claim 31, wherein the
19 state information comprises an identity of the remote computer, a network
20 endpoint identity, an identity of the remote object, and data.
21

22 **34.** A computer-readable medium as recited in claim 31, wherein the
23 object-oriented network protocol comprises a remote procedure call (RPC)
24 protocol.
25

1 **35.** A computer-readable medium as recited in claim 31, further
2 comprising computer-executable instructions that, when executed on one or more
3 processors, direct a computing device to:

4 subsequently call the remote object using the object-oriented network
5 protocol; and

6 submit the state information to the remote object.

7
8 **36.** A computer-readable medium comprising computer-executable
9 instructions that, when executed on one or more processors, direct a computing
10 device to:

11 create a state-caching object that contains state information pertaining to a
12 request for a local program object received from a remote application program
13 interface;

14 generate a reply; and

15 return the reply together with the state-caching object to the client.

16
17 **37.** A computer-readable medium as recited in claim 36, wherein the
18 state-caching object contains a computer ID field to identify a computer that
19 created the state-caching object, an endpoint ID field to identify a network
20 endpoint, an object ID field to identify the local program object, and a data field.

1 **38.** A computer-readable medium as recited in claim 36, further
2 comprising computer-executable instructions that, when executed on one or more
3 processors, direct a computing device to insert the state-caching object into a
4 context thread used to process the request.

5
6 **39.** A computer-readable medium as recited in claim 36, further
7 comprising computer-executable instructions that, when executed on one or more
8 processors, direct a computing device to:

9 subsequently receive the state-caching object in a subsequent request; and
10 use the state-caching object to recall the state information.

11
12 **40.** A computer-readable medium comprising computer-executable
13 instructions that, when executed on one or more processors, direct a computing
14 device to:

15 submit a request to a remote computer using a non-HTTP protocol over a
16 network;

17 receive a reply from the remote computer, the reply containing session state
18 information; and

19 cache the session state information for use in subsequent communication
20 with the remote computer.

21
22 **41.** A computer-readable medium as recited in claim 40, wherein the
23 session state information comprises a remote computer ID, a network endpoint ID,
24 and data.
25

1 **42.** A computer-readable medium as recited in claim 40, wherein the
2 non-HTTP protocol comprises an object-oriented network protocol.

3
4 **43.** A computer-readable medium as recited in claim 40, wherein the
5 non-HTTP protocol comprises a remote procedure call (RPC) protocol.

6
7 **44.** A computer-readable medium as recited in claim 40, further
8 comprising computer-executable instructions that, when executed on one or more
9 processors, direct a computing device to:

10 submit a subsequent request to the remote computer; and
11 send the session state information along with the subsequent request.

12
13 **45.** A computing device comprising:
14 means for calling a remote object that resides on a remote computer using
15 an object-oriented network protocol;

16 means for receiving a reply from the remote computer, the reply containing
17 state information pertaining to the remote computer; and

18 means for caching the state information for use in subsequent
19 communication with the remote computer.

20
21 **46.** A computing device comprising:
22 means for receiving a request for a local program object from a remote
23 application program interface;

24 means for creating a state-caching object that contains state information
25 pertaining to the request;

1 means for generating a reply; and
2 means for returning the reply together with the state-caching object to the
3 client.

4
5 **47.** A network comprising:
6 means for routing a request from a first computer to a second computer;
7 means for routing a reply from the second computer back to the first
8 computer, the reply carrying state information of the second computer that pertains
9 to the request; and
10 means for maintaining the state information on behalf of the first and
11 second computers.